The first two programs

01 Master Program

02 Slave Program

They're for the Arduino environment

The program called

04 VSC PIO projects

It is to use with the Visual Studio Code program plus the Platform-IO plugin

In this repository, I will publish the modifications made to the programs to control a ROV, started the programs with the Arduino IDE, and with two Arduino Mega 2560 boards, but in view of the size and complication of the codes, I am now using Visual Studio Code, with Platfor IO, and we may have to change the boards to two ESP32s, given the problems that arise with the debugging of the codes.

As it says in the title of the repository, these programs will evolve and change, as it progresses in the debugging of the codes as of the two programs.

The programs right now, compile with the Arduino IDE, but do not operate all the good that they have problems, so he put in the title of the programs, "they do not work yet", I hope to be able to solve all the problems soon.

I will keep you informed of the changes, a greeting to everyone.

The original version taken as a starting point is as follows

------------------------------------ original configuration--------------------------------------------------

ROVPS2Control\_Slavev8. ino

Hamish Trolove - 30 March 2016

[www.techmonkeybusiness.com](https://ssl.microsofttranslator.com/bv.aspx?ref=TAns&from=&to=en&a=www.techmonkeybusiness.com)

This sketch takes commands sent to you from the Master unit with the PS2 controller connected and converts it to motor commands, Servo commands, light controls, and so on. The data is sent from the Arduino (master) to the ROV (slave) using Bill Porter's EasyTransfer library via a 9600 baud (100 m anchor) serial link. The MS5803\_14 library is by Luke Miller

[http://github.com/millerlp](https://ssl.microsofttranslator.com/bv.aspx?ref=TAns&from=&to=en&a=http%3A%2F%2Fgithub.com%2Fmillerlp)

See also how the original project develops with the following codes:

HoryzonTrigger.ino, ROVPS2Control\_Masterv0.ino,

ROVDoNothing.ino, ROVSubBv0.ino, DigitalCompassv2.ino,

[PTLoggerv4.ino](https://ssl.microsofttranslator.com/bv.aspx?ref=TAns&from=&to=en&a=PTLoggerv4.ino)

and TMP36\_Temperature\_Sensor.ino.

On the

[www.techmonkeybusiness.com](https://ssl.microsofttranslator.com/bv.aspx?ref=TAns&from=&to=en&a=www.techmonkeybusiness.com)

page

Data sent from the master are raw configurations for the ESC control.

This sketch is designed for an Arduino Nano with a single serial port.

I'm using it on a Mega 2560, and it works

Pin assignments are;

D13 - RED LED pin.

D12 - Headlight Control

D11 - Bridge Pin

D8 - ESC Vertical Left

D7 - Right Vertical ESC

D6 - Horizontal LEFT ESC

D5 - Horizontal ESC Right

D4 - Camera Pass Servo (overridden in advisor codeplaza1)

D3 - Video trigger (overridden in the pro-de-order code1)

D2 - Photo Trigger (overridden in the deferred EA code1)

For the transmission of the video advisorplaza1 will use a Raspberry Pi 3 plus, a USB camera, and the program VCL Media Player

A7 - Connecting the voltage divider

A6 - TMP36 temperature sensor output pin (this sensor is changed to obsolete, DHT22 sensors are used)

GND pins of the i2c bus on MS5803-14BA and BMP180 sensors to Nano GND Pins Vcc pins on MS5803-14BA and BMP180 sensors at Nano 3.3V pins

SDA on MS5803-14BA and BMP180 sensors to Nano A4 pin sCL pins on MS5803-14BA and BMP180 sensors for Nano A5 pin

These sensors are changed to obsolete, dhT22 sensors and the MPU6050 gyroscope are used

5V - Supply to the temperature sensor TMP36. (This sensor is changed to obsolete, DHT22 sensors are used)

Serial communications connection: modified by Adviserplaza1, 26/03/2020, to separate the data sent between Arduinos, from the data to be sent to the Serial Monitor

Serial connection:

Pin 19 Arduino Control (TX) to Pin 18 Arduino ROV (RX)

Pin 18 Arduino Control (RX) to Pin 19 Arduino ROV (TX)

Connect the GND on both

Note that THE ESCs have already been scheduled at this point in the project.

Internal voltage, heading, depth, and internal temperature data are sent via the series link back to the master

For viewing on a 16x2 LCD screen.

The 16x2 LCD screen will be replaced by a basic Nextion display of 5"

The header is from an HMC5883L digital compass (i2c 0x1E address)

And the depth of a high pressure sensor MS5803-14BA (direction i2c 0x76)

(This sensor is changed to obsolete, the MPU 6050 gyroscope is used)

-----------------------------------------find the original configuration---------------------------------------------

Modified byplaza advisor1 on the different dates indicated

The first tests and code changes were initiated in 2,016, due to the number of modifications

As of now 30/01/2020, dates and changes are made to the project description

To make it easier to find errors in different code changes

30/01/2020 All the control lines of the video camera and LCD have already been commented on

I add two DHT22 and so will be the 3 that I need, one for the inner temperature of the helmet, and two for each of the protection tubes of the motors, the helmet I will remove it in due course, since with the gyroscope GY MPU6050, I have managed to measure also the temperature, therefore I will have two sensors that indicate the internal temperature of the ROV.

I've erased several control lines from the camera that won't be used

01/02/2020 I add the controls of the MPU6050 gyroscope, provided by Dea to [www.cambatronics.com](https://ssl.microsofttranslator.com/bv.aspx?ref=TAns&from=&to=en&a=www.cambatronics.com)

02/02/2020 I add the order to read only once X second the DHT22, trick by José Luis to [www.cambatronics.com](https://ssl.microsofttranslator.com/bv.aspx?ref=TAns&from=&to=en&a=www.cambatronics.com) and delete the lines corresponding to the control of the video camera, not used by asesorplaza1, in this modification of the original project

04/02/2020 I modify the reading lines of the sensors, because until now I only print in Monitor

Series the text, not the data.

06/02/2020 The code is debugging a bit, and the code header is modified.

08/02/2020 The reading error in the sensors is fixed, and the Serial Monitor is corrected.

12/02/2020 To be able to add any code, you must follow specific programming rules.

The structure of the code is modified following those programming commands, according to the information obtained and saved by me in the folder 01\_UNIR\_2\_CODIGOS.

Several blocks are created in the void loop, for the operation of each of the sensors, eliminating all delay().

13/02/2020 The DS18B20 temperature sensor is included, assisted by DEA's [www.cambatronics.com](https://ssl.microsofttranslator.com/bv.aspx?ref=TAns&from=&to=en&a=www.cambatronics.com)

16/02/2020 The E18-D80NK infrared distance sensor is added, error-free and I manage to adapt a code to read the PRESSURE sensor SKU237545

08/03/2020 A relay is connected to pin 12 of the Arduino Slave, to turn on the external light bulbs, there is no need to change the code.

26/03/2020 The MPU6050 library is included to control the gyroscope

Fixed data sending, separating data from the Serial Monitor, and the data to be sent to the Arduino Maestro, initiating two serial ports with the commands

Serial.begin(57600); // USB Serial Port, for computer communications

Serial1.begin(57600); // Serial port on pins 18 and 19, for communications with the other Arduino.

ETin.begin(details(rxdata), &Serial1); // It begins the reception of data through Serial, on pins 18 and 19, with the other Arduino.

ETout.begin(details(txdata), &Serial1); // Start sending data through Serial, on pins 18 and 19, with the other Arduino.

28/03/2020 All control instructions are included, to have all buttons operational (not working well yet)

03/04/2020 The new part of the MPU6050 Gyroscope is included, showing the 3 tilt axes, plus the temperature of the Gyroscope.

Also included is the second pressure sensor SKU237545

05/04/2020 The code of the infrared sensor E18D80NK is changed to a better one.

11/04/2020 I correct the code according to the instructions of my friend Atilano Belinchón, and the code improves quite a bit.

13/04/2020 I correct the readings and send data from the E18D80NK infrared sensor.

15/04/2020 The steering servo is included to operate it with the R1 and R2 buttons on the controller according to information obtained from the page

[https://arduparatodos.blogspot.com/2017/02/controlar-servomotor-con-arduino-varios.html](https://ssl.microsofttranslator.com/bv.aspx?ref=TAns&from=&to=en&a=https%3A%2F%2Farduparatodos.blogspot.com%2F2017%2F02%2Fcontrolar-servomotor-con-arduino-varios.html)

When pressing R1, the servo tilts the shaft where the motor tubes are located, and the ROV must rise to the surface, pressing R2, the shaft tilts in the opposite direction and the ROV must descend gaining depth.

18/04/2020 Some "if" are changed to "else if" to give the code agility, following the DEA's instructions

[www.cambatronics.com](https://ssl.microsofttranslator.com/bv.aspx?ref=TAns&from=&to=en&a=www.cambatronics.com)

Between 24/03/2020 and 28/04/2020, several code changes are made to improve it and make the controller work, following Bigbadbob's instructions in forum posts

[http://www.homebuiltrovs.com/rovforum/viewtopic.php?f=18&t=1908](https://ssl.microsofttranslator.com/bv.aspx?ref=TAns&from=&to=en&a=http%3A%2F%2Fwww.homebuiltrovs.com%2Frovforum%2Fviewtopic.php%3Ff%3D18%26t%3D1908)

From page 4

28/04/2020 Corrects spelling of all non-order text

03/05/2020 PS2 controller pin assignments are changed, according to Github code, for the Arduino Mega 2560

[https://github.com/Lynxmotion/Arduino-PS2X](https://ssl.microsofttranslator.com/bv.aspx?ref=TAns&from=&to=en&a=https%3A%2F%2Fgithub.com%2FLynxmotion%2FArduino-PS2X)

The new pin assignments are:

3.3V output to red PS2 PIN

Pin 24 to PS2 yellow PIN + 1k resistance

Pin 23 to PIN Orange PS2

Pin 25 to PIN brown PS2 + 1k resistance

Pin 22 to PS2 Blue PIN

Fixed the code to get me out in the Master the state of the spotlights, on/off

04/05/2020 Code is debugged according to Bigbadbob's advice from

[http://www.homebuiltrov.com](https://ssl.microsofttranslator.com/bv.aspx?ref=TAns&from=&to=en&a=http%3A%2F%2Fwww.homebuiltrov.com)

05/05/2020 Macro F is included, in all Serial.print, to reduce both the space occupied by the program, and the consumption of dynamic memory, according to the advice of Bigbadbob of [https://www.homebuiltrov.com](https://ssl.microsofttranslator.com/bv.aspx?ref=TAns&from=&to=en&a=https%3A%2F%2Fwww.homebuiltrov.com) , but gives me problems.

06/05/2020 Code is debugged for better performance, according to Bigbadbob's advice from [http://www.homebuiltrov.com](https://ssl.microsofttranslator.com/bv.aspx?ref=TAns&from=&to=en&a=http%3A%2F%2Fwww.homebuiltrov.com) and follows problems with Macro F

07/05/2020 Code is debugged for better performance, according to Bigbadbob's advice from [http://www.homebuiltrov.com](https://ssl.microsofttranslator.com/bv.aspx?ref=TAns&from=&to=en&a=http%3A%2F%2Fwww.homebuiltrov.com) and follows issues with Macro F

08/05/2020 Fixed Macro F issues in all fixed-text [Serial.prints](https://ssl.microsofttranslator.com/bv.aspx?ref=TAns&from=&to=en&a=Serial.prints)

You can't put Macro F, to [Serial.print](https://ssl.microsofttranslator.com/bv.aspx?ref=TAns&from=&to=en&a=Serial.print) sensor data

And it is possible to reduce the consumption of dynamic memory considerably

09/05/2020 Lines are still corrected for code debugging, according to bigbadbob's advice from [http://www.homebuiltrov.com](https://ssl.microsofttranslator.com/bv.aspx?ref=TAns&from=&to=en&a=http%3A%2F%2Fwww.homebuiltrov.com)

But the problems with the PS2 command continue.

10/05/2020 I notice an error in the order to turn on the spotlights and correct it. But the controller still doesn't work properly.

11/05/2020 Several button control commands are modified, but the controller still does not work properly.

13/05/2020 The control of the motors is modified including the ESC.h library, from the RC\_ESC\_master file, from here, instead of controlling the ESCs as a servo, is controlled as what they are ESC.

16/05/2020 Various changes are made, according to Tsanshon's advice from

[www.foromodelismonaval.es](https://ssl.microsofttranslator.com/bv.aspx?ref=TAns&from=&to=en&a=www.foromodelismonaval.es)

to expedite the code.

02/06/2020, Ignoring Tsanshon, I install the Atom program plus the Platform-IO plug-in, to debug the programs

06/06/2020, I register on the Platfor-IO forum, looking for solutions, and tell me that it is no longer supported for Atom, so I have to uninstall this program, and install Visual Studio Code, with the Platform-IO plugin

07/06/2020, After getting the programs to work in VSC + PIO, it turns out that I can not continue the work, because the components of VSC + PIO have not been updated, for Arduino, most of the people who use these programs, also use other boards, which are the ESP32, with many more possibilities than Arduino, I have already bought some plates of those, and I am waiting for them to reach me, they are cheaper than the Arduino boards.

08/06/2020 after many tests and modifications, I get the programs to compile in VSC + PIO, but I can not manage to do much more, since there is no specific library for the PS2 command.

09/06/2020 I get to introduce specific libraries with the help of Maxgerhardt de Platforio forum, according to the publications of the forum

[https://community.platformio.org/t/un-rov-de-agua-dulce-a-sweet-water-rov/14157](https://ssl.microsofttranslator.com/bv.aspx?ref=TAns&from=&to=en&a=https%3A%2F%2Fcommunity.platformio.org%2Ft%2Fun-rov-de-agua-dulce-a-sweet-water-rov%2F14157)

11/06/2020 I register on Github, open a post and upload the codes, which still do not work properly.

18/07/2020 I translate the readme file into English